Expert Systems with Applications 38 (2011) 14127–14135

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Entertainment recommender systems for group of users

Ingrid A. Christensen *, Silvia Schiaffino

ISISTAN, Facultad de Ciencias Exactas, UNCPBA, Campus Universitario, Paraje Arroyo Seco, Tandil, Argentina
CONICET, Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina

ARTICLE INFO

Keywords:
Group recommender system
Group model
Preference aggregation

ABSTRACT

Recommender systems are used to recommend potentially interesting items to users in different domains. Nowadays, there is a wide range of domains in which there is a need to offer recommendations to group of users instead of individual users. As a consequence, there is also a need to address the preferences of individual members of a group of users so as to provide suggestions for groups as a whole. Group recommender systems present a whole set of new challenges within the field of recommender systems. In this article, we present two expert recommender systems that suggest entertainment to groups of users. These systems, *jMusicGroupRecommender* and *jMoviesGroupRecommender*, suggest music and movies and utilize different methods for the generation of group recommendations: merging recommendations made for individuals, aggregation of individuals' ratings, and construction of group preference models. We also describe the results obtained when comparing different group recommendation techniques in both domains.

1. Introduction

Recommender systems appeared due to the need to provide assistance to users with a wide variety of potentially interesting items in a given domain, by identifying items that matched a user's preferences, tastes, and needs. These systems are used to give users suggestions about interesting items in different domains, such as e-commerce (Schafer, Frankowski, Herlocker, & Sen, 2007, chap. 9), movies (Miller, Albert, Lam, Konstan, & Riedl, 2003), tourism (Schiaffino & Amandi, 2009) and restaurants (Burke, Hammond, & Young, 1996), among others.

In some cases, the domain in which recommendations are originated suggests the need for personalization techniques for groups of users, instead of focusing on individual users. For example, application domains such as restaurants, TV programs (Lieberman, 1999; Yu, Zhou, Hao, & Gu, 2006), board games, movies (O'Connor, Cosley, Konstan, & Riedl, 2001), music or tourism tend to be used more frequently by groups of people rather than by individuals.

Group recommendation expands recommender systems research area (Jameson & Smyth, 2007, chap. 20); as the idea of generating a set of recommendations that satisfy a group of users with possible competing interests is a significant challenge. In relation to individual recommendations, some new issues have to be considered with group recommendations. In Jameson and Smyth (2007, chap. 20), the authors organize these issues in terms of four subtasks that must (or could) be carried out by group recommenders: obtaining information about user preferences, generating recommendations, providing explanations for these recommendations, helping users to reach consensus and a final decision. The subtask in which researchers have focused thus far is generating recommendations.

There are several algorithms or techniques for aggregating individual models or ratings that can be applied to generate recommendations to group of users, such as: merging recommendations made for individuals, aggregation of individuals' ratings for particular items, construction of group preference models, maximizing average satisfaction and minimizing misery, among others. Thus, it becomes important to evaluate and compare the different proposals in real application domains.

In this article we present two group recommender systems called *jMusicGroupRecommender* and *jMoviesGroupRecommender*. These systems were built based on a framework we developed named *GroupRecommendation*, which provides techniques to generate group recommendations focusing on the three approaches most widely used in this area: merging recommendations made for individuals, aggregation of individuals' ratings, and construction of group preference models. The goal of our work is to analyze and compare the different techniques in two different domains: music and movies.

The rest of the article is organized as follows: Section 2 presents an overview of our proposal and the techniques we will compare; Section 3 describes the music group recommender; Section 4 presents the movie group recommender; Section 5 describes the experimental results obtained when analyzing the techniques in the two

* Corresponding author at: ISISTAN, Facultad de Ciencias Exactas, UNCPBA Campus Universitario, Paraje Arroyo Seco, Tandil, Argentina.
E-mail addresses: ichriste@exa.unicen.edu.ar, christenseningrid@gmail.com (I.A. Christensen), sschia@exa.unicen.edu.ar (S. Schiaffino).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.04.221

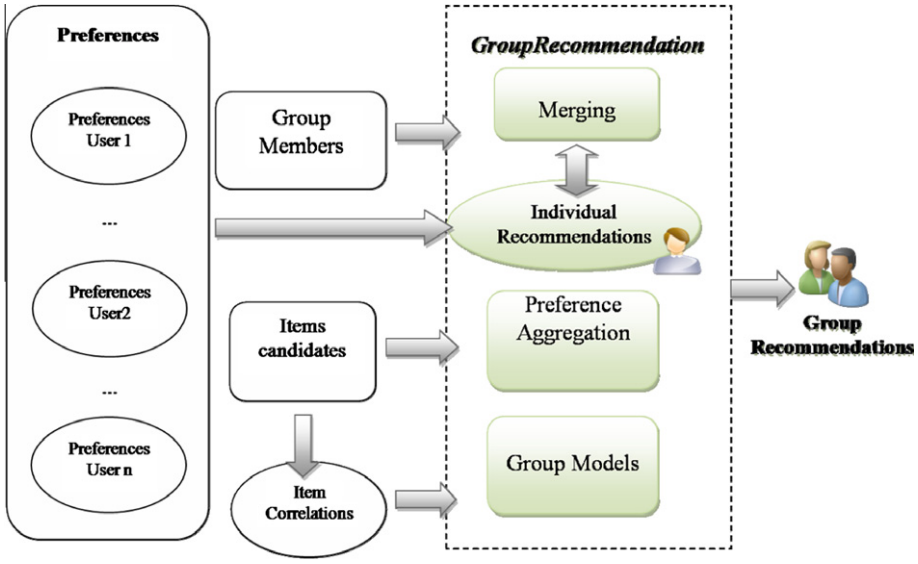

Fig. 1. General scheme of the *GroupRecommendation* framework.

entertainment domains; Section 6 presents a comparative analysis between the implemented techniques; Section 7 mentions some related works; and finally Section 8 presents our conclusions.

2. Overview of a group recommender system

In this section we will describe a Java[1] framework that provides techniques to generate recommendations within domains where items tend to be explored by groups of users rather than individuals. The framework implements the techniques that have shown, thus far, better performance in existing group recommenders: merging recommendations made for individuals; aggregation of individuals' ratings; and construction of group preference models.

GroupRecommendation is founded on different techniques that generate estimations for those items that were not explored by the group members, based on the specification of individual ratings; and then generate recommendation from these estimations. *GroupRecommendation* is easily extensible and adaptable to the needs of specific applications that want to incorporate the techniques it provides.

The main goal of this tool is to provide a set of components from which developers can build personalized recommender systems for group of users. To achieve this, the framework defines the structure upon which we can implement different group recommendation techniques, simplifying in this way the creation of new algorithms. An overview of a group recommender system built using *GroupRecommendation* can be found in Fig. 1. *GroupRecommendation* implements six techniques for group recommendation: four aggregation techniques, one merging technique, and one that defines a group model. These techniques are described in the following subsections. Additionally, we present an hybrid technique that combines two of the above approaches.

2.1. Merging of individual recommendations

A simple method for generating group recommendations is to obtain a small number of recommended solutions for each member and then merge them into a single list. Firstly, for each member (m_j), each candidate item (c_i) receives an estimative rating (r_{ij}) according to the member's profile and previous evaluations of other items than those c_i with higher estimated ratings are compile in a short list (C_i). After that, the short lists for each member (C_i) are merged into a group short list (U_jC_j) to recommend to the group as a whole.

Because this method is based on the generation of recommendations for individuals, it can also be implemented easily as an extension of an existing recommender system for individuals.

2.2. Aggregation of individual ratings

Another method for generating group recommendations is to obtain an aggregate for individual ratings for each candidate items. Firstly, for each candidate item (c_i), an estimated rating (r_{ij}) is predicted according to individual member's profiles (m_j). Then an aggregated rating (R_i) is computed from the whole set of estimated ratings $\{r_{ij}\}$. Finally, the candidate items with highest predicted ratings are recommended.

The application of this aggregation method requires the choice of particular techniques, since distinct techniques suit different desirable goals, such as total satisfactions or equity. Conflicts may arise when different goals are pursued; therefore, four of these basic aggregation techniques are described below.

2.2.1. Multiplicative

This algorithm makes an aggregate rating R_i multiplying the individuals ratings and recommends the set of candidates with the highest R_i. A disadvantage of the multiplicative strategy is that certain individual users (m_j) might always lose out, because their opinions happen to be a minority.

The aggregated rating for a group is obtained using Eq. (1):

$$R_i = \prod_{j=1}^{n} r_{ij} \tag{1}$$

2.2.2. Maximizing average satisfaction

The goal of maximizing average satisfaction can be achieved by an aggregation function (R_i) that computes some sort of average of the predicted satisfaction for each member (m_j) to be used as a basis for the selection of candidates items, as shown in Eq. (2):

$$R_i = average(\{r_{ij}\}) = \frac{1}{n} \cdot \sum_{j=1}^{n} r_{ij} \tag{2}$$

[1] http://java.sun.com/.

2.2.3. Minimizing misery

The goal of this algorithm is to minimize dissatisfaction. Even if average satisfaction among group members is high, recommended items can leave one or more members dissatisfied. Recommendations by this method are based on the lowest predicted ratings for each candidate item (c_i), favoring candidate items for which the lowest predicted rating for any group member is relatively high (see Eq. (3))

$$R_i = min_j r_{ij} \tag{3}$$

2.2.4. Ensuring some degree of fairness

The goal of this algorithm is to provide a fair degree of satisfaction to most of the group members; this goal is usually combined with some other goal, for instance the aggregation of individual predicted ratings might incorporate a penalty term that mirrors the amount of variation among the predicted rating.

The equation for this algorithm (Eq. (4)) was combined with the algorithm for maximizing average satisfaction. In the second term, w represents the relative importance of fairness

$$R_i = average(\{r_{ij}\}) - w \cdot standard\ deviation \tag{4}$$

2.3. Construction of group preference models

The goal of this method is to obtain a group model that represents the preferences of the group as a whole and generate recommendations taking into account this group profile through known recommendation techniques for individuals, such as collaborative filtering or demographic profile.

The following steps describe the process to generate recommendation constructing a profile for groups as a whole:

1. Construction of a preference model M that represents the preferences of the whole group.
2. For each candidate item c_i, M is used to predict the rating R_i for the whole group.
3. Recommendation of the set of candidate items with the highest predicted rating R_i.

There are even more possible methods for the construction of group models than for the aggregation of individual rating, because group profiles can take many different forms.

In this work, this method was implemented on the basis of the average resulting of the combination of individual member's profiles.

2.4. Hybrid technique

This technique combines merging of individual recommendation and multiplicative aggregation. Merging is used to filter candidate items, taking only those with the highest rating for each individual member. Then, an aggregate rating is obtained multiplying individual ratings for those short listed candidate items.

3. Music recommendation to groups of users

The music recommender system, *jMusicGroupRecommender*, was built on top of an application that manages and plays music files, also developed by us (Christensen, Bugarini, Casamayor, & Schiaffino, 2008). This application enables users to visualize, index, catalog, add and organize files in a library using different criteria, as well as to obtain different statistics about the use of the application.

Fig. 2 is a snapshot of the application that shows a playlist example with the different attributes of the songs.

The original system provided recommendations to individual users using collaborative filtering and considering the similarities among the different songs belonging to different users' libraries. To enable the system to make group recommendations, we implemented the four subtasks a group recommender system requires: obtaining information about user preferences, generating recommendations, providing explanations for these recommendations, helping users to reach consensus and a final decision.

Fig. 2. Snapshot of music recommender system.

Fig. 3. Qualification of the tracks by users in *jMusicGroupRecommender*.

In order to obtain information about users' preferences, this system applies two techniques: explicit specification of preferences and implicit detection. Group members can rate the different tracks in a library, indicating whether or not they liked a track, as shown in Fig. 3. Those unevaluated tracks are rated as "undefined". On the other hand, the implicit detection of individual users' preferences is done in two ways:

1. When users are reproducing files the system considers skipped "undefined" tracks as "disliked". When a track is played for longer than a predefined threshold, the system considers the track as "liked".
2. The system considers the number of times different tracks are played. Thus, a track that was reproduced many times more than another with the same assessment by the user, will have a higher rating.

For both individual recommendations and groups recommendations, the system generates a data model based on the preferences created by the user and their interaction with the system. This data model maintains the system-generated rating for each item-user relationship.

The system can perform estimates for unrated tracks with the help of this data model and the correlations among the items. These estimates are used to make recommendations to groups of people. The framework generates the estimates for unrated tracks based on the correlation between different items, using the Eq. (5)

$$predictedRating_{i,u} = \frac{\sum_{j=0}^{n}(correlation_{ij} * specifiedRating_{ju})}{\sum_{j=0}^{n} correlation_{ij}} \quad (5)$$

Item similarities are assessed to generate (individual or group) recommendations. The correlation between two different songs was obtained by computing the similarities of the different attributes describing both songs. The system instantiates the different techniques implemented in the framework to obtain the group recommendation, allowing the selection and comparison in this domain. Fig. 4 shows the generation of recommendations through the selection of the algorithm.

The system gives the option to limit the result list to tracks that were never played by any member of the group, i.e., the system recommends only tracks unexplored by group members. The system allows the user to select the number of recommendations in the final list. This option may be interesting in the music domain, because group members can determine the amount of songs, for example, in relation to playing time.

Next to the recommendation list, the music recommender system provides an explanation for the suggestions, showing the attributes of the different tracks that were considered in the similarity

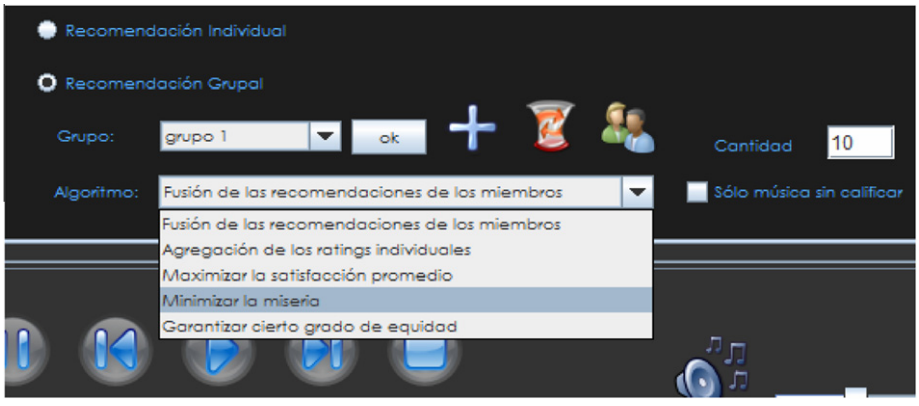

Fig. 4. Selection of group recommendation technique.

Fig. 5. Qualification of the movies by users in *jMoviesGroupRecommender*.

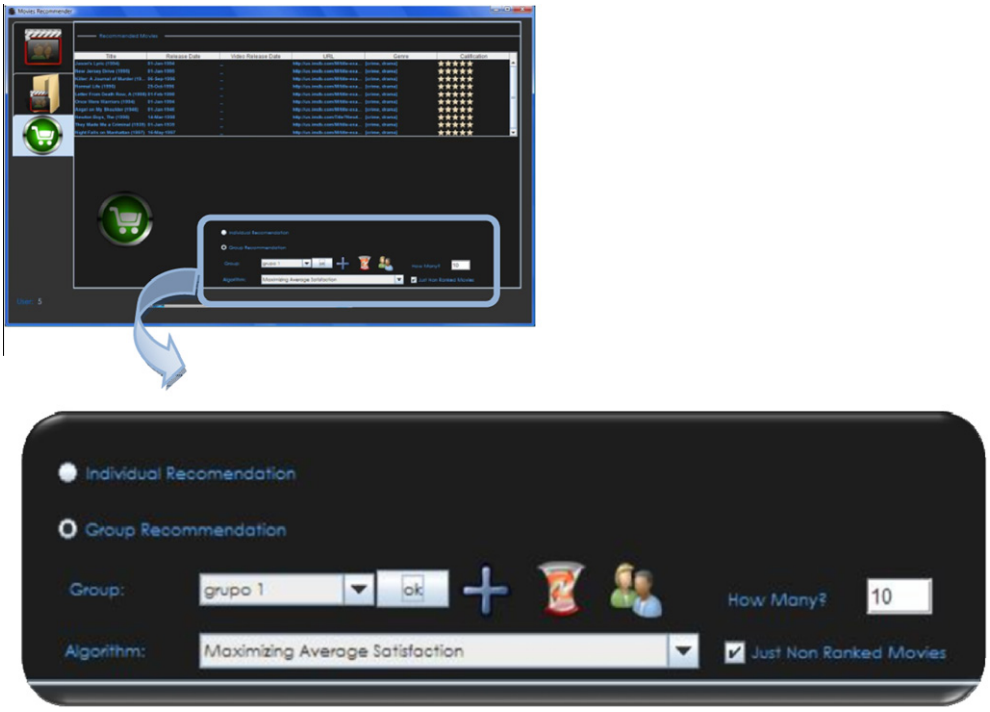

Fig. 6. Group recommendation in *jMoviesGroupRecommender*.

calculus, and the rating given by the user that requests the recommendation for the whole group.

4. Movie recommendation to group of users

The movie recommender system enables users to handle movies by varying attributes such as title, directors, actors, release date, genres and URL address where movie information can be found, among others. In each user's session the system displays a list of movies rated by any user with its corresponding rating; a list of all the movies contained in the system, which the user can rate; and a list of movies recommended for that user. The system also enables the creation of group of users in order to recommend movies to these groups.

To obtain information about each user's preferences the system utilizes explicit feedback, since the system can recommend movies but not play them. Thus, the system trusts the user's ratings. No implicit feedback can be obtained.

At this point, the system trusts on the users' evaluations, because it assumes that the use of the system is only in order to obtain acceptable and consistent recommendations. The range of qualifying varies between 1 and 5 stars (see Fig. 5).

The correlation between two different movies was obtained by computing the similarities of the different attributes describing both movies. The system takes this correlation into account to generate individual recommendations and rating estimations to generate a list of recommendations for a given group. The most important attribute when calculating the correlation between movies is the genre/s.

On the other hand, to generate recommendations the system uses the *GroupRecommendation* framework, enabling the selection of any of the implemented algorithms (see Fig. 6).

Each user of the system has definite preferences for certain movies. The recommender system maintains all the preferences and generates a data model, which is combined with the correlation between the movies to obtain the estimates for unrated movies. These estimates are used to carry out the various recommendations (individual or group).

The system is responsible for the administration of user groups. Each user can create his/her own groups including other system users, who may observe in his/her session all the groups which were included. Thus, all group members can request for group recommendations at any time.

To obtain the group recommendation the users must:

1. Select the group of users.
2. Select the group recommendation technique.
3. Specify if the result is limited to movies that were never watched by any member of the group.
4. Specify the number of recommendations.

As regards the explanation of recommendations, the system displays a list of results showing the attributes that influenced the calculus of the correlation, together with the rating the active user gave to the movie.

5. Experimental results

In order to compare and analyze the techniques implemented within the *GroupRecommendation* framework it was necessary to experiment using extensive datasets for both domains: music and movies. Yahoo! Webscope Program[2] was particularly appropriate as it contains numerous users' preferences for a wide range of candidate items.

Ten experiments were designed for each of the domains considered in the present work so as to analyze the behavior of the different recommendation techniques. Each experiment was carried out following this procedure:

1. Obtain a certain number of users from the dataset.
2. Create a group with the users selected.
3. Generate recommendations for the group varying the recommendation technique.

[2] http://research.yahoo.com/Academic_Relations.

4. Calculate the individual member satisfaction for each of the group members.

In step 4, it was necessary to implement some criteria to compare and analyze the techniques; therefore individual member satisfaction was considered in all the experiments.

The result of each experiment is a list of recommended items for a single group; the list obtained in each experiment had 10 recommended items (tracks or movies), which were then used as a basis for the comparative analysis, obtaining the normalized individual satisfaction by dividing the sum of ratings of the recommended items in the list by the maximal "possible" sum for each individual member.

The groups of users in each experiment were randomly selected, because it is supposed that a group of people who want to watch a movie or listen to music together may not have similar preferences or tastes.

These experiments were carried out to analyze the behavior of the alternative techniques in different application domains and groups of users with varied profiles, and obtain a comparative analysis based on the normalized individual satisfaction for each group member.

A detailed explanation of the twenty experiments would exceed the scope of an article, therefore only two of the experiments carried out in each domain and their results are described in the following subsections. The overall results include the results of all the experiments.

5.1. Experiments in the music domain

The dataset used to carry out the experiments with the music recommender were provided by Yahoo! and represent a snapshot of the preferences of the Yahoo! Music community with regard to a vast collection of tracks. The dataset contains over 717 million ratings for 138 thousand music tracks given by 1.8 million users of Yahoo! Music services.

The data were collected between 2002 and 2006. Each song in the dataset is accompanied by certain attributes (artist, album and genre) which are used by the system to evaluate the correlation between the songs to predict the ratings for the songs unevaluated by the users.

5.1.1. Experiment 1

In this experiment four profiles of randomly selected users from the dataset were utilized. The group members for the analysis are the following:

- User 2: Profile with 358 songs evaluated.
- User 4: Profile with 70 songs evaluated.
- User 5: Profile with 21 songs evaluated.
- User 6: Profile with 1350 songs evaluated.

The system predicts the unspecified preferences for the members taking into account the users' profiles.

Fig. 7 shows the values of the normalized individual satisfaction for this group of users.

In this experiment, algorithms corresponding to the aggregation of the individual preferences approach obtain the highest normalized individual satisfactions for each member of the group. While merging of individual recommendations obtain the lowest normalized individual satisfaction. The hybrid technique generally achieves high normalized individual satisfactions, and sometimes even the highest.

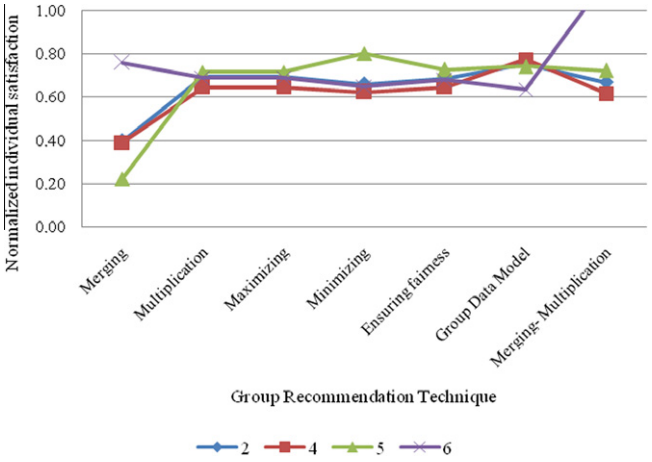

Fig. 7. Normalized individual satisfaction of the members of Group 1 for each group recommendation technique.

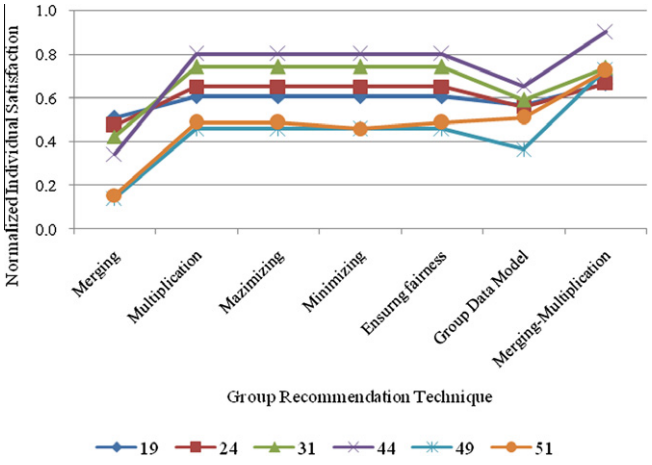

Fig. 8. Normalized individual satisfaction of the members of Group 2 for each group recommendation technique.

5.1.2. Experiment 2

In this experiment six profiles of randomly selected users from the dataset were utilized. The group members for the analysis are the following:

- User 19: Profile with 349 songs evaluated.
- User 24: Profile with 533 songs evaluated.
- User 31: Profile with 54 songs evaluated.
- User 44: Profile with 328 songs evaluated.
- User 49: Profile with 197 songs evaluated.
- User 51: Profile with 169 songs evaluated.

The system predicts the unspecified preferences for the group members taking into account their individual profiles. Fig. 8 shows the normalized individual satisfaction values for this group of users.

In this experiment, merging of individual recommendation obtained the lowest normalized individual satisfaction; whereas the hybrid technique achieves the highest individual satisfaction in all the instances.

5.1.3. Overall results in the music domain

Fig. 9 shows the satisfaction achieved for each group members created in the different experiments. On the *x*-axis we can see all

the group recommendation techniques implemented in the framework and used in both music and movie recommender. On the *y*-axis we have the values of normalized individual satisfaction.

In the experiments we obtained the normalized individual satisfaction for each user who belongs to one of the groups used in the different experiments.

The graph shows two important things. First the reliability offered by each technique, which is given by the dispersion of the intersection points. Secondly, noting the highest satisfaction values obtained may determine the performance of each technique.

The dispersion points on the graph show the reliability of the technique studied because if there are large variations between individual satisfactions of the group members analyzed, the technique does not guarantee satisfaction equality of all members.

The ideal is minimal dispersion and achieve high levels of satisfaction for all members, which would mean that the technique is reliable.

5.2. Experiments in the movie domain

The dataset used to carry out these experiments with the movies recommender were provided by Yahoo! and represent a snapshot of the preferences of the Yahoo! Movies community in relation to a vast collection of movies. This dataset contains detailed information about numerous movies released prior to November 2003, including cast, crew, synopsis, genre, average ratings, awards, among others.

Some of these attributes, such as genre, directors, crews, actors and title, were used to analyze the correlation between two movies.

5.2.1. Experiment 1

Four profiles of randomly selected users were used in this experiment.

- User 9: Profile with 11 movies evaluated.
- User 5: Profile with 23 movies evaluated.
- User 15: Profile with 50 movies evaluated.
- User 26: Profile with 11 movies evaluated.

The system predicts the unspecified preferences for the group members taking into account their individual profiles.

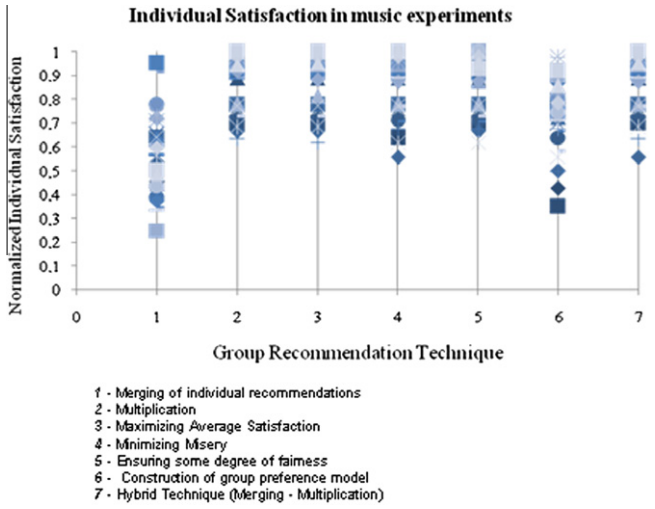

Fig. 9. Normalized individual satisfaction for users of all groups used in the 10 experiments.

Fig. 10 shows the normalized individual satisfaction values for this group.

In this experiment, the algorithm for constructing a group preference model obtained the lowest normalized individual satisfaction.

5.2.2. Experiment 2

Seven profiles of randomly selected users were used in this experiment. The group members for the analysis are the following:

- User 3: Profile with 73 movies evaluated.
- User 4: Profile with 18 movies evaluated.
- User 5: Profile with 12 movies evaluated.
- User 6: Profile with 45 movies evaluated.
- User 7: Profile with 19 movies evaluated.
- User 8: Profile with 19 movies evaluated.
- User 9: Profile with 11 movies evaluated.

The system predicts the unspecified preferences for the group members taking into account their individual profiles.

Fig. 11 shows the normalized individual satisfaction values for this group.

As in the previous experiment, the algorithm for constructing a group preference model obtained the lowest normalized individual satisfaction. Merging of individual recommendation acquired low values of individual satisfactions too.

5.2.3. Overall results in the movie domain

Fig. 12 shows the results obtained in the 10 experiments in the movie domain. The graphic exhibits all the normalized individual satisfaction obtained for each group member in the experiments. On the *x*-axis we can see all the group recommendation techniques implemented in the framework and used in both music and movie recommender. On the *y*-axis we have the values of normalized individual satisfaction.

The merging of individual recommendation present a large variation between the individual satisfactions of the group members, because each item satisfies a single member without securing anything in the rest of the group.

In this domain, aggregation and hybrid techniques presented higher values of satisfaction with minimal dispersion.

6. Discussion and analysis

In this work we presented a comparative analysis between different group recommendation techniques based on three

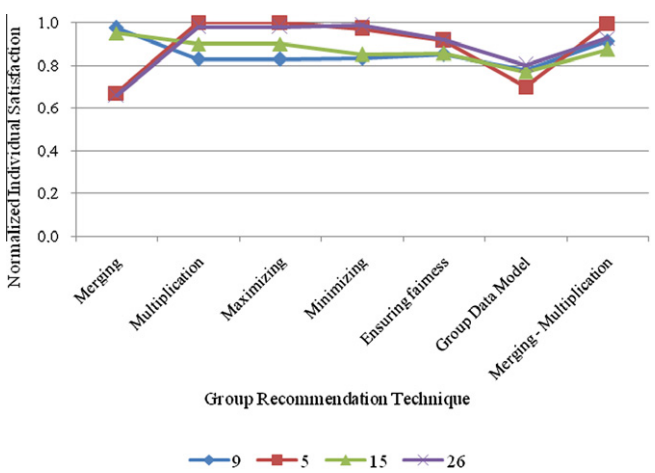

Fig. 10. Normalized individual satisfaction of the members of Group 1 for each group recommendation technique.

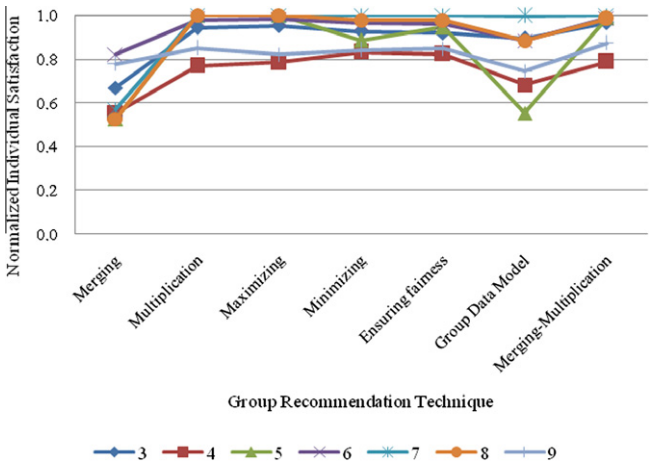

Fig. 11. Normalized individual satisfaction of the members of Group 2 for each group recommendation technique.

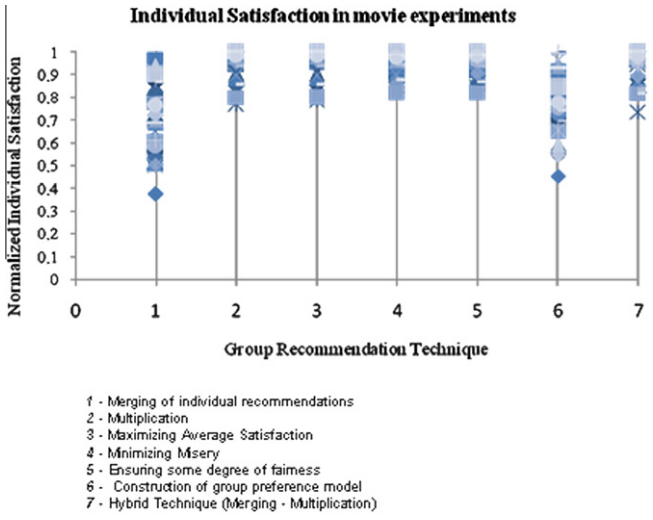

Fig. 12. Normalized individual satisfaction for users of all groups used in the 10 experiments.

well-known approaches: merging of the individual recommendation, aggregation of individual ratings and constructing a group model. We implemented and compared a merging technique; four aggregation techniques (multiplication, maximizing of average satisfaction, minimizing misery and ensuring some degree of fairness); one technique for constructing a model group and a hybrid technique created from the merging technique and multiplicative aggregation.

The experiments were carried out with different groups of users selected from the dataset provided by Yahoo! Webscope Program. In each case, the created groups differ in the number of users and size of each member's profile.

The results obtained in those experiments reflect the behavior of the different techniques. For instance, merging of individual recommendations achieved low levels of individual members' satisfaction in the diverse analyzed groups; which was expected, taking into account that this algorithm obtains a short list that satisfies each individual member and then creates a unique result list merging the individual recommendations. This technique ensures that a portion of resulting group list satisfies each member, but not the rest of the group members.

On the other hand, the results obtained with the algorithm designed to create a group profile through the average of individual

Table 1
Comparison of the different group recommender systems.

Domain	Polylens (O'Connor et al., 2001)	Travel Decision Forum (Jameson, 2004)	MusicFX (McCarthy & Anagnost, 1998)	*jMusicGroupRecommender*	*jMoviesGroupRecommender*
	Movies	Tourism	Music	Music	Movies
Subtasks					
1. Obtaining information about user preferences	Explicit specification	Explicit and collaborative specification	Explicit specification	Explicit and implicit specification	Explicit specification
2. Generating recommendations	Merging users' recommendations combining with least misery	Group model	Group model	Merging users' recommendations/multiplication/maximizing average/minimizing mysery/ensuring fairness/group model	Merging users' recommendations/multiplication/maximizing average/minimizing mysery/ensuring fairness/group model
3. Providing explanations	Showing predictions of individuals	Averaging and aggregation	Not supported	Showing predicted rating and attributes of the recommended tracks	Showing predicted rating and attributes of the recommended movies
4. Helping users to reach consensus and a final decision	Not supported	Animated characters that represent each member group	Not supported	Not supported	Not supported

preferences, exposed that this technique lacks accuracy in the interpretation of the preferences of the group as a whole; because, in some cases, the taste of a single member was generalized to the rest of the group. Because of this problem, this technique obtained low values of individual satisfaction in all experiments. The aggregation algorithms (multiplication, maximizing of average satisfaction, minimizing misery and ensuring fairness) obtained acceptable results in all the experiments with fairly equitable levels of individual satisfaction among the different group members and achieved high individual satisfaction values. Each aggregation algorithm has a specific goal, for instance, minimizing misery avoids solutions that leave one or more members highly dissatisfied. In some cases, because of the eagerness to achieve these goals, some algorithms satisfy some members, leaving the rest of the group completely dissatisfied; or, in the case of the ensuring fairness algorithm, achieving fairness among members could reduce all the individuals' satisfaction.

Finally, the hybrid technique, based on merging and multiplication techniques, obtained high individual satisfaction values for each individual member in all the experiments. This is because the merging technique selects the best candidate items for each individual member and then applies the aggregation technique, maximizing the multiplication of individual ratings.

7. Related works

There are some recommender systems that try to adapt their results to the requirements of group recommendation, applying specific techniques to obtain information about group members' preferences and to aggregate these preferences to generate recommendations. For example, Adaptive Radio (Chao, Balthrop, & Forrest, 2004) focuses on what members dislike to obtain their preferences. On the other hand, Travel Decision Forum (Jameson, 2004; Jameson, Baldes, & Kleinbauer, 2004) uses a technique that shares user preferences, which can reduce effort, enable learning from other members' preferences and summarize preferences.

As regards recommendation techniques, Intrigue (Ardissono, Goy, Petrone, Segnan, & Torasso, 2003) uses aggregation of individual ratings by defining a function that considers the number of users in a sub-group and the importance of the sub-group (children and people with disabilities for example are considered as more important). Pocket Restaurant Finder (McCarthy, 2002) and Fit (Goren-Bar & Glinansky, 2002) apply a variant of maximizing average satisfaction. Polylens (O'Connor et al., 2001) uses the minimizing misery strategy. Let's Browse (Lieberman, 1999) builds a group preference model making a linear combination of individual models. Other examples of group recommender systems can be found in Jameson and Smyth (2007, chap. 20).

In Baskin and Krishnamurthi (2009) the authors present a preference aggregation algorithm which considers only the relative preferences. This algorithm uses the preferences to extract a partial ordering of items from each reviewer's score data, and aggregates these orderings.

All the systems mentioned above utilize only one technique to generate recommendations. Using *GroupRecommendation*, we can build recommender systems that can choose the recommendation strategy to be used, as *jMoviesGroupRecommender* and *jMusicGroupRecommender* do.

Table 1 summarizes the information about the different systems we studied.

8. Conclusions

We have presented in this work two entertainment recommender systems for group of users. We have evaluated these systems and we have presented a comparative analysis of the performance of different group recommendation techniques, such as merging of individual recommendations, aggregation of individual ratings, and construction of group preference models. From the results we have obtained, other developers of group recommender systems can decide which technique to apply, according to the goal they want to fulfill when making recommendations.

As a future work, we are planning to analyze the techniques implemented in *GroupRecommendation* in other application domains, such as tourism. We are also planning to extend the framework functionality by implementing new recommendation techniques and supporting the subtasks of obtaining user preferences, explaining recommendations, and aiding users to reach a final decision.

Acknowledgement

This work was partially supported by ANPCyT (Argentina) through PICT 2007 Project No. 529.

References

Ardissono, L., Goy, A., Petrone, G., Segnan, G., & Torasso, G. (2003). Intrigue: Personalized recommendation of tourist attractions for desktop and handset devices. In *Applied artificial intelligence* (pp. 687–714). Taylor and Francis.

Baskin, J. P., & Krishnamurthi, S. (2009). Preference aggregation in group recommender systems for committee decision-making. In *RecSys '09: Proceedings of the third ACM conference on recommender systems* (pp. 337–340). New York, NY, USA: ACM.

Burke, R. D., Hammond, K. J., & Young, B. C. (1996). Knowledge-based navigation of complex information spaces. In *Proceedings of the 13th national conference on artificial intelligence* (pp. 462–468). AAAI Press.

Chao, D. L., Balthrop, J., & Forrest, S. (2004). Adaptive radio: Achieving consensus using negative preferences. In *Proceedings of the 2005 international ACM SIGGROUP conference on supporting group work, Sanibel Island, Florida, United States* (pp. 120–123).

Christensen, I., Bugarini, F., Casamayor, A., & Schiaffino, S. (2008). Recomendación personalizada de música. In *Proceedings of the III Congresso da Academia Trinacional de Ciencias, C3N. Foz do Iguacu, Brasil: C3N*. ISSN 1982-2758.

Goren-Bar, D., & Glinansky, D. (2002). Family stereotyping – A model to filter TV programs for multiple viewers. In *Proceedings of the 2nd workshop on personalization in future TV*.

Jameson, A. (2004). More than the sum of its members: Challenges for group recommender systems. In *proceedings of the working conference on advanced visual interfaces (AVI '04)* (pp. 48-54). New York, NY, USA: ACM. ISBN 1-58113-867-9.

Jameson, A., & Smyth, B. (2007). Recommendation to groups. In *The adaptive web: Methods and strategies of web personalization* (pp. 596–627).

Jameson, A., Baldes, S., & Kleinbauer, T. (2004). Two methods for enhancing mutual awareness in a group recommender system. In *Proceedings of the international working conference on advanced visual interfaces*.

Lieberman, H., Van Dyke, N., & Vivacqua, A. (1999). Let's browse: A collaborative web browsing agent. In *Intelligent user interfaces* (Vol.12, pp. 65-68).

McCarthy, J. F. (2002). Pocket restaurant finder: A situated recommender system for groups. In *Proceedings of the workshop on mobile ad-hoc communication at the 2002 ACM conference on human factors in computer systems*. Minneapolis: ACM.

McCarthy, J. F., & Anagnost, T. D. (1998). Musicfx: An arbiter of group preferences for computer supported collaborative workouts. In *proceedings of the ACM conference on computer supported cooperative work* (pp. 363-372).

Miller, B. N., Albert, I., Lam, S. K., Konstan, J. A., & Riedl, J. (2003). Movielens unplugged: Experiences with an occasionally connected recommender system. In *Proceedings of ACM 2003 conference on intelligent user interfaces (IUI'03) (accepted poster)*. Chapel Hill, North Carolina: ACM.

O'Connor, M., Cosley, D., Konstan, J. A., & Riedl, J. (2001). Polylens: A recommender system for groups of users. In *ECSCW'01: Proceedings of the seventh conference on European conference on computer supported cooperative work* (pp. 199–218). Norwell, MA, USA: Kluwer Academic Publishers.

Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In *The adaptive web: Methods and strategies of web personalization* (pp. 291–324).

Schiaffino, S. N., & Amandi, A. (2009). Building an expert travel agent as a software agent. *Expert Systems with Applications, 36*, 1291–1299.

Yahoo! Academic Relations (2002–2006). R4 – Yahoo! movies user ratings of movies and movie descriptive content information, version 1.0. <http://research.yahoo.com/Academic_Relations>.

Yahoo! Academic Relations (2003). R2 – Yahoo! music user ratings of songs with artist, album, and genre meta information, version 1.0. <http://research.yahoo.com/Academic_Relations>.

Yu, Z., Zhou, X., Hao, Y., & Gu, J. (2006). TV program recommendation for multiple viewers based on user profile merging. *User Modeling and User-Adapted Interaction, 16*, 63–82.